Progress (100%)

Item 1/30

Which of the following sentences is true?

```
nums = [1, 2, 3]
vals = nums
```

- `vals` is longer than `nums`
- `nums` is longer than `vals`
- `nums` and `vals` are different lists
- (●) `nums` and `vals` are different names of the same list

Next →

Retake Test | Back to Summary

Progress (100%)

Item 2/30

What is the output of the following snippet?

```
def fun(x):
    if x % 2 == 0:
        return 1
    else:
        return 2

print(fun(fun(2)))
```

- the code will cause a runtime error
- `1`
- `2` (selected)
- `2None`

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 3/30

What is the output of the following piece of code?

```
x = 1
y = 2
x, y, z = x, x, y
z, y, z = x, y, z

print(x, y, z)
```

- 1 2 2
- **1 1 2**
- 1 2 1
- 2 1 2

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 4/30

Take a look at the snippet and choose the true statement:

```
nums = [1, 2, 3]
vals = nums
del vals[:]
```

- `vals` is longer than `nums`
- `nums` and `vals` are different lists
- **(selected)** `nums` and `vals` are different names of the same list
- the snippet will cause a runtime error
- `nums` is longer than `vals`

← Prev
Next →

Progress (100%)

Item 5/30

The result of the following division:

```
1 // 2
```

- is equal to `0.0`
- **is equal to `0`**
- is equal to `0.5`
- cannot be predicted

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 6/30

What is the output of the following snippet?

```
dct = { 'one':'two', 'three':'one', 'two':
v = dct['three']

for k in range(len(dct)):
    v = dct[v]

print(v)
```

- ('one', 'two', 'three')
- **one**
- two
- three

← Prev Next →

Retake Test Back to Summary

Progress (100%)

Item 7/30

One of the following variable names is illegal - which one?

- `in_`
- `in`
- `IN`
- `In`

← Prev | Next →

Retake Test | Back to Summary

Progress (100%)

Item 8/30

What is the output of the following snippet?

```
def fun(x, y):
    if x == y:
        return x
    else:
        return fun(x, y-1)

print(fun(0, 3))
```

- 2
- 1
- 0
- the snippet will cause a runtime error

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 9/30

What is the output of the following snippet?

```
def fun(inp=2, out=3):
    return inp * out
print(fun(out=2))
```

- 2
- the snippet is erroneous
- 4
- 6

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 10/30

Assuming that the `tuple` is a correctly created tuple, the fact that tuples are immutable means that the following instruction:

```
tuple[1] = tuple[1] + tuple[0]
```

- (●) is illegal
- ( ) may be illegal if the tuple contains strings
- ( ) is fully correct
- ( ) can be executed if and only if the tuple contains at least two elements

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 11/30

What is the output of the following snippet?

```
list =  [x * x for x in range(5)]
def fun(lst):
    del lst[lst[2]]
    return lst

print(fun(list))
```

- [ ] `[0, 1, 4, 16]`
- [ ] `[0, 1, 9, 16]`
- [x] `[0, 1, 4, 9]`
- [ ] `[1, 4, 9, 16]`

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 12/30

What is the output of the following piece of code?

```
x = 1 // 5 + 1 / 5
print(x)
```

- 0
- 0.0
- 0.2
- 0.4

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 13/30

How many elements does the `lst` list contain?

```
lst = [i for i in range(-1, -2)]
```

- three
- one
- zero
- two

← Prev | Next →

Retake Test | Back to Summary

Progress (100%)

Item 14/30

What is the output of the following snippet?

```
dct = {}
dct['1'] = (1, 2)
dct['2'] = (2, 1)

for x in dct.keys():
    print(dct[x][1],end="")
```

- (2,1)
- (1,2)
- 12
- 21

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 15/30

What is the output of the following piece of code if the user enters two lines containing `2` and `4` respectively?

```
x = float(input())
y = float(input())
print(y ** (1 / x))
```

- 0.0
- 4.0
- **2.0**
- 1.0

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 16/30

How many hashes (`#`) will the following snippet send to the console?

```
lst = [[x for x in range(3)] for y in rang

for r in range(3):
    for c in range(3):
        if lst[r][c] % 2 != 0:
            print("#")
```

- six
- nine
- three
- zero

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 17/30

What is the output of the following piece of code if the user enters two lines containing `3` and `6` respectively?

```
y = input()
x = input()
print(x + y)
```

- (•) 63
- ( ) 36
- ( ) 3
- ( ) 6

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 18/30

The following snippet:

```
def func(a, b):
    return b ** a

print(func(b=2, 2))
```

- will output `None`
- is erroneous
- will output `4`
- will output `2`

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 19/30

What is the output of the following piece of code?

```
print("a", "b", "c", sep="sep")
```

- abc
- a b c
- asepbsepcsep
- **asepbsepc**

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 20/30

What value will be assigned to the `x` variable?

```
z = 0
y = 10
x = y < z and z > y or y > z and z < y
```

- False
- 0
- **True**
- 1

← Prev | Next →

Retake Test | Back to Summary

Progress (100%)

Item 21/30

The meaning of a *positional argument* is determined by:

- its value
- its position within the argument list
- its connection with existing variables
- the argument's name specified along with its value

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 22/30

What is the output of the following snippet?

```
tup = (1, 2, 4, 8)
tup = tup[-2:-1]
tup = tup[-1]
print(tup)
```

- (•) `4`
- ( ) `(4,)`
- ( ) `(4)`
- ( ) `44`

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 23/30

The following snippet:

```
def func1(a):
    return None

def func2(a):
    return func1(a) * func1(a)

print(func2(2))
```

- will output `2`
- will output `16`
- **will cause a runtime error**
- will output `4`

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 24/30

What is the output of the following snippet?

```
dd = { "1":"0", "0":"1" }
for x in dd.vals():
    print(x, end="")
```

- `1 0`
- `0 1`
- the code is erroneous (the `dict` object has no `vals()` method)
- `0 0`

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 25/30

How many stars (`*`) will the following snippet send to the console?

```
i = 0
while i < i + 2 :
    i += 1
    print("*")
else:
    print("*")
```

- zero
- two
- one
- **the snippet will enter an infinite loop, printing one star per line**

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 26/30

What will be the output of the following snippet?

```
a = 1
b = 0
a = a ^ b
b = a ^ b
a = a ^ b

print(a, b)
```

- 1 0
- **0 1**
- 0 0
- 1 1

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 27/30

An operator able to check whether two values are not equal is coded as:

- `not ==`
- `!=`
- `<>`
- `=/=`

← Prev Next →

Retake Test Back to Summary

Progress (100%)

Item 28/30

What is the output of the following piece of code if the user enters two lines containing `3` and `2` respectively?

```
x = int(input())
y = int(input())
x = x % y
x = x % y
y = y % x
print(y)
```

- (•) 0
- ( ) 3
- ( ) 2
- ( ) 1

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 29/30

What is the output of the following snippet?

```
lst = [1, 2]

for v in range(2):
    lst.insert(-1, lst[v])

print(lst)
```

- [ ] `[1, 2, 1, 2]`
- [x] `[1, 1, 1, 2]`
- [ ] `[2, 1, 1, 2]`
- [ ] `[1, 2, 2, 2]`

← Prev
Next →

Retake Test
Back to Summary

Progress (100%)

Item 30/30

Which of the following lines incorrectly invokes the function defined as:

```
def fun(a, b, c=0):
```

- `fun(0, 1, 2)`
- `fun(b=1)`
- `fun(a=0, b=0)`
- `fun(a=1, b=0, c=0)`

← Prev

Retake Test
Back to Summary